(19) JAPANESE PATENT OFFICE (JP)

(12) PUBLICATION OF UNEXAMINED PATENT APPLICATION (A)

(11) Kokai (Unexamined) Patent Publication Number: 58-80740

(43) Date of Disclosure: May 30, 1984

(51) Int. Cl.^{3} Identif. Symbol Intra-Agency Number
G 06 G 9/22 8120-5B

Examination requested: not yet requested
Number of Inventions: 1 (total of 4 pages)

(54) Title of the Invention: MICROPROGRAM CONTROLLER

(21) Application Number: 56-179316
(22) Application Date: November 9, 1981

(72) Inventor: Masanobu WATANABE
c/o NEC Corporation
Tokyo-to, Minato-ku
Shiba Go-chome, 33-ban, 1-go

(71) Applicant: NEC Corporation
Tokyo-to, Minato-ku
Shiba Go-chome, 33-ban, 1-go

(74) Representative: Susumu UCHIHARA, patent attorney.

SPECIFICATIONS

1. Title of the Invention: Microprogram Controller

2. Scope of the Patent's Claims

A microprogram controller, characterized by the fact that in an arithmetic operation system which is controlled by a microprogram transferring data to an arithmetic part, having a register part comprising a plurality of registers, indicated registers are selected from the content of a counter which is controlled by an instruction of said microprogram.

Detailed explanation of the Invention

This invention relates to an information processing device which is controlled by a

microprogram controller.

Registers that were used in the past for information processing systems were generally constructed as registers that were using 1 or 2 stacks, or multiple registers which could be used with direct stack operations were employed. To make it possible to realize stack operations with this method, these dedicated stack registers had to use the memory of the main storage unit, which means that the data always had to be exchanged between the central processing unit and the main storage device.

In information processing systems used in recent years, on the other hand, very fast stack operations must be realized with mathematical formulas such as reverse Polish notation for various structural elements.

However, because the number of registers required for stack calculations is smaller than the number of dedicated stack registers with the register methods which use only stacks as described above, the problem is that the memory used by the main memory unit is insufficient. Because of that, the problem was that it was not possible to achieve smooth performance of stack processing operations.

In view of the problems related to prior art as described above, the purpose of this invention is to provide a microprogram controller which makes it possible to realize effective processing of stack calculating operations by either avoiding transfer of data between the central processing device and the main memory, which is required for stack calculating operations, or by reducing as much as possible the number of transfers.

[page 2]

According to this invention, a microprogram controller is obtained, characterized by the fact that in an arithmetic operation system which is controlled by a microprogram transferring data to an arithmetic part, having a register part comprising a plurality of registers, indicated registers are selected from the content of a counter which is controlled by an instruction of said microprogram.

The following is an explanation of the present invention with reference to the enclosed figures.

Figure 1 is a construction diagram showing one embodiment of this invention. The construction of the microprogram controller of this invention comprises a control part 1, a register part 2, and an arithmetic part 3. The control part 1 comprises a microprogram controlling part 10, a microprogram instruction decoding part 11, and a counter 12, while the construction of the register part 2 comprises selection parts 21 ~ 23, and registers 24 ~ 26.

A summary of the microprogram controller of this invention will be provided first.

When the microprogram instruction decoding part 11 fetches a microprogram instruction 19 from the microprogram memory part 10, this instruction is processed with arithmetic processing relating to the register content, and it will be determined from the instruction field 111 of a microprogram instruction 19 whether an instruction has been issued performing register selection (this instruction will be hereinafter called a selection arithmetic instruction) depending on the content of the counter 12. If this instruction 19 is a register selection arithmetic operation instruction, either the B field 113 of the microprogram instruction 19 will be selected, or the value of the base register specified with the D field 114 will be added to the number of the counter 12 by an adder 13, and registers 24 ~ 26 will be selected based on this value. The determination as to which of the items B field 113 and D field 114 is to be fetched can be made based on the content of the instruction field 111. Moreover, the specifications of the register selection can be performed with the A field 112 independently from the content of the counter 12.

When the value of the base register has been fetched from the B field 113, a selection part B22 selects the content of the register, which is determined by content of the counter 12 and of the B field 113, as the input for arithmetic processing with the arithmetic part 3. On the other hand, when the value of the base register data is fetched from the D field, a selection part D21 will select from the content of the counter 21 and the D filed 114 and set output data required for an arithmetic result by the arithmetic part 3. Specifically, because a selection part A 23 and a selection part B22 transfer to the arithmetic part 3 data read from the content of the specified registers 24 ~ 26, the selection part D 21 will set the resulting data of the arithmetic part 3 to the specified registers 24 ~ 26.

The value of the counter 12 is controlled by a counter update field 115. Specifically, the value of the counter update field 115 is created as a new counter value added to the value of the counter 12 at this time.

The explanation above was a simplified explanation of a microprogram controller of this invention.

A more detailed explanation will now be provided to explain the manner in which the stack arithmetic operations are performed with this microprogram controller.

The stack arithmetic operation will be explained first with reference to Figures 2 (a), (b), (c), and (d).

The stack arithmetic operation consists of a pop operation, in which the header of the stack at the present point is fetched (the operation in which data X is fetched in Figure 2 (c) is calculated in Figure 2 (d)), and of a push operation, in which data X in Figure 2 (b) is added in front of Y in Figure 2 (c) for newly added data of the header of the stack at the present point. These pop operations and push operations can be realized with the microprogram controller of

this invention as follows.

First, during the initial state, 0 is set to the counter 12, and because the base register is specified with the B field 113 and with the D field 114, since the register 24 is set as the register number 1, 0 will be set both to the B field 113 and to the D field 114 (0 = register number 1). At this time, the push instruction adding to the stack of the present data Z will set 0 to the D field 114, and 1 will be set to the counter update field 115. When this instruction is received, first, the value of the D field 114 (0) will be added by a D bus 54_1, and the counter value (1) of the counter update field 115 will be added and transferred to the adder 14 by a counter value fetching line 56_1.

[page 3]

The result of the addition of the adder 14 (1 = 0 + 1) is transferred through the D bus 54_2 to a selection part D21. The selection part D21 selects the register 24 based on this value (1) and data Z is obtained through the D bus 54_3 from the selection part 3 and set to this register part 24 (creating the status shown in Figure 2 (a).

Similarly, when 1 is set to the counter updated field 115, per each sequential addition of the value 1 of the counter 12, that is to say when the value 2 is changed to 3, data Y and data X is set with the push instruction to respective registers 25, 26, and after the status shown in Figure 2 (b), the stack shown in Figure 2 (c) will be formed. A content of the counter 12 corresponding to 3 is created at this point. Next, a pop instruction fetching data X for the header of this stack will set 0 to the B field 113, and 1 will be set to the counter update field 115. Specifically, first, the adder 13 receives from the B bus 53_1 the value (0) of the B field 113, and the value (3) of the B field 113 is received from the counter value fetching line 56_2 and the result of this addition is transferred to the selection part B 22.

Next, the selection part B22 will select register 26 based on the value (3) received from the adder 13, and the content of the register 26 (data X) will be transferred through the B bus 53_3 to the adder 3. After that, the value (-1) of the stack update field 115 is added to the value (3) of the counter 12, and the value of the counter is updated to 2 (= 3 – 1). This resulting value is shown in Figure 2 (d). The value of the counter 12 is updated with a different timing with the counter update field 115 according to the microprogram instructions with the push instruction and with the pop instruction. Specifically, while with the push instruction, the content of the counter 12 is updated by the counter update field 115 before the selection register is set, updating of the counter value is performed after the determination of the register with the pop instruction.

According to this invention, because all the registers can be used for stack arithmetic operations passing through the counter 12 which is used for register selection, and because the pop arithmetic operation and the push arithmetic operations can be realized with one instruction, the data processing can be performed efficiently and with a high degree of efficacy during the stack arithmetic operation.

In addition, this invention can be also easily configured in the same manner with the construction of a microprogram controller having n registers.

Brief Explanation of Figures

Figure 1 is a construction diagram showing one embodiment of this invention, and Figures 2 (a), (b), (c) and (d) are diagrams explaining the stack arithmetic operation of this invention.

The numbers in these figures indicate: 1 ... controller, 2 ... register part, 3 ... arithmetic part, 12 ... counter, 19 .... microprocessor instruction, 24, 25, 26 ... registers.

Representative: Susumu UCHIDA, patent attorney [personal seal].

Figure 1

[page 4]

Figure 2 (a), (b), (c), (d)